

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

REMARKS

Applicant is in receipt of the Office Action mailed January 28, 2004. Claims 9, 39, 61, 76, 81, and 85 have been amended to more clearly claim the invention. Claims 1-7, 9-37, 39-59, 61-74, and 76-90 remain pending in the case. Reconsideration of the present case is earnestly requested in light of the following remarks.

Re: Telephone Interview with Examiner

In the telephone interview with the Examiner, summarized above, the Examiner asserted that new claim 90 was not supported in the Application, specifically the limitation "wherein said automatically generating the graphical program comprises automatically generating graphical code in the graphical program without direct user input." Applicant respectfully disagrees.

The originally filed application included claims 9, 39, 61, and 76 which each included the following limitation:

"wherein said automatically generating the graphical program comprises automatically generating graphical code in the graphical program without direct user input."

Applicant thus respectfully submits that claim 90 is fully supported in the Application.

In the telephone interview, the Examiner also asserted that no support was provided in the Application for the added limitation of the amended independent claims, specifically, the limitation "wherein said automatically generating the graphical program comprises automatically including the nodes in the graphical program, wherein said automatically including the nodes in the graphical program is performed without direct user input selecting the nodes", which is included in independent claims 1, 31, and 71. Applicant respectfully disagrees.

Applicant submits that there are numerous passages in the Application that support this limitation. For example, page 23, lines 22-26 reads:

“As described below, an API which enables a client program to programmatically generate or edit a graphical program may be supported by a graphical programming system. Such an API may be utilized by a prototyping environment application to enable the automatic generation of a graphical program to implement a prototype, *without requiring the type of user interaction described above with reference to Figures 4 – 5.*”

Figures 4-5 describe interactive or manual creation of a graphical program, as opposed to automatic generation of a graphical program. For example, the user interaction described with reference to Figures 4-5 includes the following descriptions:

From page 22, lines 3-4:

“In step 422 of Figure 4, the user adds other objects/nodes to or edits other objects/nodes of the graphical program.”

From page 22, lines 16-18:

“The user may add nodes and other types of objects to a graphical program in various ways, e.g., by selecting a node or object from a palette that displays icons representing the various nodes and objects.”

and

From page 22, lines 19-20:

In step 424 of Figure 4, the user connects or “wires” the graphical program objects in order to achieve the desired executable logic and data flow or control flow.

Thus, Applicant submits that the Specification clearly supports the feature that the automatic generation of the graphical program does not include the user selecting and adding nodes to the graphical program and the user connecting the nodes in the graphical program.

Additionally, further support for this limitation is provided by the following:

As shown above, the originally filed claims 9, 39, 61, and 76 each included the limitation:

“wherein said automatically generating the graphical program comprises automatically generating graphical code in the graphical program without direct user input.”

Originally filed claim 25 includes the limitations:

“wherein said automatically generating the graphical program comprises:

generating portions of graphical code, wherein each portion of graphical code implements one of the functions;

linking the portions of graphical code together.”

Originally filed claim 26 includes the limitations:

“wherein each portion of graphical code includes one or more graphical program nodes, wherein each node has one or more inputs or outputs;

wherein generating each portion of graphical code comprises connecting the node inputs and outputs together in order to implement the function with which the portion of graphical code is associated.”

Additionally, page 59, line 25-26, in the description of Figure 43 reads:

“When this program is run, the graphical code shown within the while loop executes until the input to the condition terminal 612 becomes ‘false’.” Thus, as Figure 43 shows, the graphical code within the while loop comprises interconnected nodes.

Thus, “graphical code” is clearly defined as nodes for use in the graphical program. Moreover, automatic generation of the graphical program is clearly described in the various claims as automatically generating graphical code in the graphical program without direct user input, where automatically generating the graphical program includes generating portions of graphical code, wherein each portion of graphical code implements one of the functions, and where each portion of graphical code includes one or more graphical program nodes, where each node has one or more inputs or outputs, and linking the portions of graphical code together, where generating each portion of graphical code comprises connecting the node inputs and outputs together in order to implement the function with which the portion of graphical code is associated.

Thus, Applicant respectfully submits that the limitation "wherein said automatically generating the graphical program comprises automatically generating graphical code in the graphical program without direct user input" is fully supported in the present application. Applicant thus submits that independent claims 53, 81, and 90 are patentably distinct over the cited art, and thus, claims 53, 81, and 90, and claims respectively dependent thereon, are allowable.

Applicant further submits that the limitation "wherein said automatically generating the graphical program comprises automatically including the nodes in the graphical program, wherein said automatically including the nodes in the graphical program is performed without direct user input selecting the nodes", included in independent claims 1, 31, and 71, is fully supported in the present application.

Since the Examiner has admitted that with this limitation, the independent claims distinguish over the cited art, Applicant submits that independent claims 1, 31, and 71, and the claims respectively dependent thereon, are patentably distinct over the cited art, and are thus allowable for at least the reasons presented above.

Additional arguments for the allowability of the present claims are presented below:

Section 102 Rejections

Claims 1-16, 21-43, 45-65 and 67-89 were rejected under §102(b) as being anticipated by U.S. Patent No. 5,862,372 to Morris et al ("Morris"). This rejection is respectfully traversed. Applicant notes that claims 8, 38, 60, and 75 were previously cancelled, and so their rejection is rendered moot.

Claim 1 recites:

1. (Previously Presented) A method of creating a graphical program to perform an algorithm, the method comprising:

recording one or more functions in response to user input, wherein the one or more functions specify the algorithm; and

automatically generating the graphical program in response to the recorded one or more functions, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, wherein the graphical program implements the algorithm;

wherein said automatically generating the graphical program comprises automatically including the nodes in the graphical program, wherein said automatically including the nodes in the graphical program is performed without direct user input selecting the nodes.

As previously argued, the present application is directed toward a method for programmatically or automatically creating a graphical program in response to functions that have been specified by a user. As one example, the user may select various image processing functions to be performed on an image, and these user-selected image processing functions may be recorded in memory as they are selected or performed. When the user has finished selecting functions to be performed on the image (and presumably the desired result has been achieved on the image), a software program may operate to programmatically generate a graphical program that implements these functions (i.e., an image processing algorithm that includes the user selected functions in the order in which the user selected them). This example is discussed in the specification with respect to Figures 6 – 14.

The present application describes a conventional prior art method for creating a graphical program with respect to Figures 4, 5A and 5B. As discussed, the conventional prior art method for creating a graphical program involves the user selecting graphical program nodes or objects, e.g., by dragging them from a palette into a block diagram window, and then interconnecting these nodes or icons together. This technique is described in greater detail in, for example, U.S. Patent No. 4,901,221, among others. Applicant submits that the Morris, et al. patent, at most, teaches this prior art technique.

With respect to the “recording step” the Office Action states that Morris discloses “recording functions in response to user input” citing column 2 lines 20-27. The cited portion of Morris refers to prior art systems which involve selection and interconnection

of logical operators or icons to create a script. Thus, the cited portion of Morris involves manually creating a block diagram or graphical program.

With respect to the "automatically generating step" the Office Action cites the Morris patent at column 3 lines 29-35. Here, the Morris patent refers to a user directly selecting and dragging icons on to the screen in a certain order to specify a sequence of operations or "script". For example, Morris states at column 3 line 32 that "Icons representing the objects (and accordingly their functionalities) may be placed (dragged) into an appropriate view. The system of the invention generates the information needed by the run time program." The Morris et al. patent further describes this operation at col. 5 lines 13 - 22:

During the development of an application with the system of this invention, the user directs the incorporation of objects into one of the four views, modifies the properties associated with that object, and interconnects the objects so chosen. . . . As the system responds to the user's directions, it creates a script representing the application being developed by the user. When the development process is complete, the user stores the script on one of the system's storage devices.

One of the interesting features of the object oriented authoring system of this invention is that no traditional looking code, written in a programming language, is generated by the system. Rather, the output of the authoring system is a listing of the objects in order of appearance (in the program sequence) along with the properties associated with that object at that place in the sequence. The properties are the information or settings which specify to the object how it will perform. The computer implemented application development system records the settings for the objects and the order of execution of the objects in an output script. Thus, the system operates only at the object level. As the author proceeds in developing the application, the system adds to the script.

Thus the Morris patent describes operation where a user manually selects objects or icons and interconnects them together to specify a script.

Claim 1 refers to "automatically generating a graphical program". This automatically generating comprises programmatically generating the graphical program, e.g., where a software program analyzes the recorded functions and then automatically generates the graphical program based on these recorded functions. The cited portion of Morris is comparable to the prior art method of a user manually creating graphical programs as described in Figures 4, 5A and 5B of the present application, (i.e., the user

manually adding objects or icons to a graphical program and manually connecting them in a desired way). The teaching of Morris is analogous to this prior art method of manually creating a graphical program, except that a script is then created based on the sequence of objects that the user selects and interconnects.

In the Morris patent, the user manually assembles a block diagram (as shown in Figure 2 of Morris) in order to specify a script. In the system of the present application, the user first provides input to create a prototype or script, and then a software program executes to programmatically generate a graphical program which implements the script. Thus, the method of the present application is virtually the opposite of that taught in Morris. Applicant notes that the programmatically generated script in Morris is *not* a graphical program. Significantly, Morris does not teach or suggest any type of programmatic generation of a graphical program. Rather, Morris teaches that a user is required to manually assemble a block diagram.

Thus, Applicant submits that Morris simply does not teach or suggest "automatically generating a graphical program". The cited portion of Morris refers to the user actually dragging the objects into the appropriate view, which is not an automatic creation of a graphical program, but rather is a manual creation. This distinction is emphasized by the further limitations of claim 1 that "the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program", and that "automatically generating the graphical program comprises automatically including the nodes in the graphical program, wherein said automatically including the nodes in the graphical program is performed without direct user input selecting the nodes."

Similarly, Applicant submits that Morris simply does not teach or suggest the limitation of independent claims 53, 81, and 90 that reads "wherein, in automatically generating the graphical program, the program instructions are executable to automatically generate graphical code in the graphical program without direct user input".

In addition, the two cited portions of Morris (one from the Background section and one from the Summary of the Invention section) which allegedly teach the recording

step and the automatically generating step both involve selecting or dragging objects or modules into an application under development. Applicant submits that these cited portions do not represent any type of coherent method that could be argued to teach or suggest the automatic or programmatic graphical program generation of the present claims.

Thus, Applicant submits that the various independent claims, and those dependent thereon, are allowable. Removal of the section 102 rejection of claims 1-16, 21-43, 45-65 and 67-89 is respectfully requested.

Section 103 Rejections

Claims 17-20, 44, and 66 were rejected under §103(a) as being unpatentable over Morris and U.S. Patent No. 5,623,659 to Shi et al. ("Shi"). This rejection is respectfully traversed.

Applicant submits that since independent claims 1, 31, 53, 71, and 81 have been shown to be allowable, dependent claims 17-20, 44, and 66 are also allowable, for at least the reasons provided above.

Additionally, as the Examiner is certainly aware, to establish a prima facie obviousness of a claimed invention, all claim limitations must be taught or suggested by the prior art. In re Royka, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974), MPEP 2143.03. Obviousness cannot be established by combining or modifying the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion or incentive to do so. In re Bond, 910 F. 2d 81, 834, 15 USPQ2d 1566, 1568 (Fed. Cir. 1990). In addition, the showing of a suggestion, teaching, or motivation to combine prior teachings "must be clear and particular Broad conclusory statements regarding the teaching of multiple references, standing alone, are not 'evidence'." In re Dembiczak, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999). The art must fairly teach or suggest to one to make the specific combination as claimed. *That one achieves an improved result by making such a combination is no more than hindsight without an initial suggestion to make the combination.*

The Office Action states, "The locking mechanism disclosed by Shi could prevent mistakes from occurring in Morris' graphical program, and laying out some control mechanisms for user manipulation. One skilled in the art would be motivated to learn from Shi's teaching by locking the association between the script and the graphical program to have more control over the user's manipulations of the graphical program".

Applicant notes that neither Morris nor Shi provides a motivation to combine, and respectfully submits that the Examiner has simply noted an improved result of the combination without an initial suggestion from the prior art to make the combination, and has thus used hindsight analysis in an attempt to reproduce Applicant's invention as claimed. Applicant thus submits that the combination of Morris and Shi is improper.

Applicant further notes that even if Morris and Shi were properly combinable, which Applicant argues they are not, the resulting combination would still not teach Applicant's invention as claimed. For example, as noted above, Applicant submits that Morris teaches manual creation of a block diagram, and programmatic generation of a script or text-based program based on the block diagram, and specifically does *not* teach programmatic generation of a graphical program as described and claimed in the independent claims. Shi also fails to teach this feature of the invention as represented in the independent claims. Thus, neither Morris nor Shi, either singly or in combination, teaches all of the features and limitations of the independent claims, and so claims 1, 31, 53, 71, and 81, and claims respectively dependent thereon, are patentably distinct and non-obvious over Morris and Shi, and are thus allowable.

Removal of the section 103 rejection of claims 17-24, 44, and 66 is respectfully requested.

Applicant also asserts that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the independent claims have been shown to be patentably distinct, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-44300/JCH.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

Respectfully submitted,



Jeffrey C. Hood
Reg. No. 35,198
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: 5/5/2004